Please Amend the Claims as Follows:

1. (Currently Amended) A main memory simulator for simulating large computer memories, wherein said large computer memories are defined by a plurality of target memory addresses, ~~wherein each of said plurality of addresses contains data, and wherein said large computer memories are large enough to prevent simulation via the use of one-to-one memory to file addressing for all of said plurality of memory addresses, wherein~~ said simulator comprising ~~comprises~~:

one or more mass storage devices having page addresses to simulate said target memory addresses;

a memory cache; and

a processor operable with said memory cache, wherein said processor operates under instructions

to move data contained in a predetermined range of frequently used target memory addresses between corresponding page addresses in said mass storage devices ~~of said plurality of memory addresses to~~ and said memory cache on a fast memory access basis, and

~~under instructions~~ to move data contained in a predetermined range of infrequently used target memory addresses between corresponding page addresses in said mass storage devices ~~of said plurality of memory addresses to~~ and said memory cache on a slow memory access basis.

2. (Currently Amended) The simulator of claim 1, wherein said fast memory access comprises the utilization of a set of fast lookup tables to directly obtain a page address that has been allocated to an address within the frequently used range of target memory addresses ~~address~~.

3. (Currently Amended) The simulator of claim 1, wherein said slow memory access comprises the utilization of a slow lookup table to first determine if a page address has been allocated to an address within the infrequently used range of target memory ~~address~~ addresses and to then obtain the allocated page address.

4. (Currently Amended) The simulator of claim 2, wherein said slow memory access comprises the utilization of a slow lookup table to determine if a page address has been allocated to an address within the infrequently used range of target memory ~~address~~ addresses and the obtainment of said page address if allocated.

5. (Original) The simulator of claim 1, wherein the movement of data is achieved through a page transfer.

6. (Currently Amended) A main memory simulator for simulating large computer memories, wherein said large computer memories are defined by a plurality of target memory addresses, ~~wherein each of said plurality of addresses contains data, and wherein said large computer memories are large enough to prevent simulation via the use of one-to-one memory to file addressing for all of said plurality of memory addresses, wherein~~ said simulator comprises:

mass storage means for simulating said target memory addresses;

means for storing data;

means for processing instructions

~~, wherein said means for processing instructions is in communication with said means for storing data; and~~

~~means for providing instructions to said means for processing instructions, wherein said means for providing instructions provides the instruction to transfer data by identifying a memory address of said data, wherein said memory address comprises at least one of said plurality of memory addresses;~~

~~wherein said means for processing instructions processes said instruction~~ to transfer data, ~~by determining if said memory address of said data is a frequently or infrequently accessed memory address, and by transferring~~ said data being transferred between said mass storage means ~~to~~ and said means for storing via a fast memory access scheme if said data ~~is~~ resides within a predetermined range of frequently accessed target memory addresses or ~~address or by transferring said data to said means for storing~~ via a slow memory access scheme if said data resides within a predetermined range of ~~is an~~ infrequently addressed target memory ~~address~~addresses.

7. (Currently Amended) The simulator of claim 6, wherein said fast memory access scheme comprises utilizing a set of fast lookup tables.

8. (Currently Amended) The simulator of claim 7, wherein said fast lookup tables enable said means for processing to directly obtain a page address in said mass storage means corresponding to said data stored within said range of frequently accessed target addresses~~address~~.

9. (Original) The simulator of claim 6, wherein said slow memory access scheme comprises utilizing a slow lookup table.

10. (Currently Amended) The simulator of claim 9, wherein said slow lookup table enables said means for processing to first determine if a page address has been allocated to said data residing within said range of infrequently addressed memory addresses~~address~~ and then ~~obtaining~~ to obtain the allocated page address.

11. (Original) The simulator of claim 6, wherein said transferring of data is achieved through a page transfer.

12. (Currently Amended) A method for simulating large computer memories via one or more storage devices, wherein said large computer memories are defined by a plurality of target memory addresses, ~~wherein each of said plurality of addresses contains data,~~ and wherein said large computer memories are large enough to prevent simulation via the use of one-to-one addressing for all of said plurality of memory addresses, the method comprising:

obtaining a request for transfer of data residing within one of said target memory addresses ~~within at least one of said plurality memory addresses, wherein said request identifies the memory address of said at least one of said plurality of memory addresses containing said data~~;

~~determining whether the requested memory address is a frequently requested memory address or whether the requested memory is an infrequently requested memory address;~~

~~if the determination reveals~~ if said data resides within a predetermined range of ~~a~~ frequently requested target memory ~~address~~addresses, then using a fast memory access scheme to ~~transfer~~ do at least one of transferring the data to, and from, a corresponding address within said storage devices~~within said frequently requested memory address~~; and

~~if the determination reveals~~ if the data resides within a predetermined range of ~~an~~ infrequently requested target memory addresses ~~address~~, then using a slow memory access scheme to do at least one of transferring ~~transfer~~ the data ~~within said infrequently requested memory address~~ to, and from, a corresponding address within said storage devices.

13. (Original) The method of claim 12, wherein said fast memory access scheme comprises utilizing a set of fast lookup tables.

14. (Currently Amended) The method of claim 13, wherein said fast lookup tables enable direct obtainment of a page address in said storage devices corresponding to a target memory address within said range of said frequently accessed target memory addressesaddress.

15. (Original) The method of claim 12, wherein said slow memory access scheme comprises utilizing a slow lookup table.

16. (Currently Amended) The method of claim 15, wherein said slow lookup table first enables determining if a page address in said storage devices has been allocated to a memory address within said range of infrequently addressed target memory address addresses that stores said data and then enables obtaining the allocated page address.

17. (Original) The method of claim 12, wherein said transfer of data comprises a page transfer.

18. (Currently Amended) A memory simulation system for simulating a main memory of a computer, wherein the spaces of memory within said main memory are defined by a main memory address, the system comprising:

a plurality of files, wherein said files include a fast look-up table and a slow look-up table,

wherein said fast look-up table is operable to directly obtain a page address that has been allocated to simulate said a corresponding main memory address that is within at least one of a predetermined range of frequently accessed main memory addresses, and

wherein said slow look-up table is operable ~~to first determine if a page address has been allocated to said main memory address and if a page address has been allocated to said main memory address,~~ to obtain ~~the allocated~~ a page address that has been allocated to simulate a corresponding main memory address that is within at least one of a predetermined range of infrequently accessed main memory addresses;

a cache~~, wherein said cache includes a buffer~~; and

an interface~~, wherein said interface receives~~ to receive a request for transfer of data residing at a requested main memory address, ~~main memory~~ said transfer occurring between at least one of said plurality of files and said cache via use of one of said fast look-up table and said slow lookup table ~~buffer, wherein said request is made through input of said main memory address to said interface, and wherein in response to said request said interface performs a page transfer between said at least one of said plurality of files and said buffer according to the page address that has been allocated to said main memory address via said fast look-up table or said slow look-up table~~.

19. (Currently Amended) The system of claim 18, wherein said plurality of files further include a last access look-up table, wherein said last access look-up table includes a last main memory address accessed and ~~the~~a page address allocated to said last main memory address accessed.

20. (Currently Amended) The system of claim 18, wherein ~~upon said interface performing a page transfer,~~ if said requested main memory address is within said at least one predetermined range of infrequently used addresses, a list of addresses simulating said main memory addresses is searched in attempt to locate a page address allocated to said requested main memory address, and if said page address is not located, allocating said page address for said requested main memory address~~represented by the page is used to search a breakpoint list, and wherein upon~~

~~finding that a word within said page has a breakpoint set as indicated by the search of said breakpoint list, a breakpoint flag within a page descriptor of said page is set.~~

21. (Original) The system of claim 18, wherein both said fast look-up table and said slow look-up table are savable into a finite number of files.

22. (Original) The system of claim 21, wherein said fast look-up table and said slow look-up table are restorable from the saved files enabling a previously stopped simulation to continue.

23. (Currently Amended) The system of claim 18, wherein said fast look-up table is used to obtain a page address for any main memory address that resides within a predetermined lowest address range within main memory address space ~~for memory that is accessed frequently and wherein said slow look-up table is used to obtain a page address for memory that is accessed infrequently~~.

24. (Currently Amended) The system of claim 23, wherein said fast look-up table is divided into a plurality of banks, each bank having a defined number of words, and wherein ~~the frequency of access is determined by dividing~~ said requested main memory address is divided by said defined number of words to obtain a quotient value, wherein said quotient value is compared against a predetermined value~~, and wherein the comparison provides an indication of~~ to determine whether said requested main memory address is within said at least one of said predetermined frequently or said infrequently accessed main memory address range.

25. (Currently Amended) A memory simulation system for simulating main memory of a computer, ~~wherein the spaces of memory within said main memory are defined by a main memory address,~~ the system comprising:

means for receiving a data transfer request to transfer data between one or more storage devices simulating said main memory, wherein said data transfer request is defined by a main memory address within said main memory;

means for determining the frequency of use of said main memory address based on which one of multiple pre-defined address ranges within said main memory contains said main memory address;

means for obtaining a page address corresponding to said main memory address, wherein said means for obtaining includes:

means for obtaining ~~a~~said page address corresponding to said main memory address when said main memory address has been determined to be frequently used; and

means for obtaining ~~a~~ said page address corresponding to said main memory address when said main memory address has been determined to be infrequently used; and

means for transferring data between said page address of one of said storage devices and an address within a different one of said storage devices ~~to a memory location, wherein said transferring of data comprises a page transfer in accordance with the obtained page address~~.

26. (Currently Amended) The system of claim 25, wherein said means for obtaining the page address ~~of the frequently used~~ when said main memory address is frequently used comprises a set of fast lookup tables.

27. (Original) The system of claim 26, wherein said fast lookup tables enable said means for obtaining to directly obtain the page address of the frequently used main memory address.

28. (Currently Amended) The system of claim 25, wherein said means for obtaining the page address ~~of the infrequently used~~ when said main memory address is infrequently used comprises a slow lookup table.

29. (Original) The system of claim 28, wherein said slow lookup table enables said means for obtaining to first determine if a page address has been allocated to said main memory address and then to obtain that allocated pages address.

30. (Currently Amended) A method of memory transfer for use in simulating a main memory~~, wherein the spaces of memory within said main memory are defined by a main memory address,~~ the method comprising:

obtaining a main memory address, wherein said main memory address indicates a request for a main memory transfer between a file simulating the main memory and a cache buffer;

determining, based on which of multiple pre-defined address ranges in said main memory contains said main memory address, whether ~~if~~ said main memory address comprises memory that is accessed frequently or infrequently;

if said main memory address comprises memory that is accessed frequently, directly obtaining a page address that has been allocated to said main memory address through use of a first look-up table;

if said main memory address comprises memory that is accessed infrequently, first determining whether a page address has been allocated to said main memory address through use of a second lookup table, then obtaining the page address that has been allocated to said main memory address;

transferring ~~the requested main memory~~ data between said page address of said file and said cache buffer ~~via a page transfer that utilizes the page address allocated to said main memory address~~.

31. (Original) The method of claim 30, further comprising the step of saving the last main memory address accessed and the corresponding page address.

32. (Currently Amended) The method of claim 30, ~~further comprising the step of searching a breakpoint list,~~ wherein said second look-up table contains a list of infrequently used ones of said main memory addresses for which page addresses have been allocated~~upon discovering that a word within the transferred page has a breakpoint set as indicated by the search of said breakpoint list, setting a breakpoint flag within a page descriptor of the transferred page~~.

33. (Original) The method of claim 30, further comprising the step of saving said first look-up table and said second look-up table into a number of finite files.

34. (Original) The method of claim 33, further comprising the step of restoring said first look-up table and said second look-up table to continue a previously stopped simulation.

35. (Original) The method of claim 30, wherein said first look-up table is divided into a plurality of banks, each bank having a defined number of words, and wherein said step of determining is achieved by dividing said main memory address by said defined number of words to obtain a quotient value, then comparing said quotient value against a predetermined value, wherein the comparison provides an indication of whether said main memory address is frequently or infrequently addressed.